Service Discovery with SWE-ET and DIANE

A Comparative Evaluation By Means of Solutions to a Common Scenario

Ulrich Küster
Compared Solutions

- SWE-ET
  (Politecnico Milano + CEFRIEL, Italy)
  - Andrea Turati, Dario Cerizza, Emanuele Della Valle, Marco Brambilla, Stefano Ceri, Federico Facca, Christina Tziviskou
  - GLUE, WSMO compliant discovery engine
  - WebML/WebRatio, Web Application Framework

- DIANE
  (University Jena, Germany)
  - Ulrich Küster, Birgitta König-Ries
  - DIANE Project (services in ad-hoc networks)
Formalism Used to Model Ontologies

- Glue based on F-logic
- Compliant with WSMO
- High expressivity
- DE (DIANE Elements) & DSD (DIANE Service Descriptions), own language specifically tailored for service descriptions
- Ontology language very lightweight (subset of F-logic without rules and quantifiers)
- Limited expressivity

→ Modeled ontologies very similar (scenarios did not require complexity)
Formalism Used to Model Goals and Services

- Web services and goals modeled with F-logic
- Follows WSMO modeling approach
- Assumptions and post-conditions
- Separates between classes and instances of goals and web services
- Set of effects an offer can provide
- Set of effects a requester accepts
- Sets comprised by
  - type condition
  - direct conditions
  - property conditions
- Fuzzy sets in requests
  - fuzzy conditions
  - connecting strategies
- Variables for service IO

⇒ Modelling of services quite different!
Goals and Services – Some Pros and Cons

- Need to rely on WebRatio for notion of current time (not supported by F-logic)
- Direct support for rules and arithmetic computations (shipping price)
- Preferences out of scope
- Custom Java code can be injected in instances (DateTime \textit{now})
- No support for rules (evaluation delegated to external service)
- Directly supports preferences

\begin{itemize}
\item \textbf{Different strengths and weaknesses}
\end{itemize}
The Process of Matchmaking - Reasoning

- Exploits standard reasoning tools for F-logic (Flora-2)
- Matching defined by domain specific wgMediator (set of F-logic rules)
- Matching done in levels (determined by rules that match)
- Custom reasoning operation offer *subset* of request?
- Generic matchmaking rules
- Configures offer in order to maximize subset
- Recursive graph matching algorithm
- Result of matchmaking: configured offers, degree of match $\in [0,1]$

→ Standard vs. custom reasoning, generic vs. domain specific matching
The Process of Matchmaking – Selection

- Views discovery and selection as separate tasks
- Glue does not support selection, left to Application
- Limited support for ranking through discrete levels

- Selection viewed as integral part of matchmaking (project goal: complete automation)
- User preferences captured through fuzzy request sets
- Ranking by degree of match

⇒ Different views on the scope of matchmaking
Dynamic Aspects of Service Descriptions

- Separates between:
  - discovery capabilities
  - negotiation capabilities
  - selection capabilities
- Negotiation supported by WebRatio

- Estimation operations (inquire information from service endpoint)
- Executed during matchmaking only if necessary (static information insufficient to guarantee match or fail)

➔ Different view on scope of matchmaking
Service Invocation

- Supported through WebRatio
- Supported by DIANE framework
- Lowering/lifting through mapping rules in service grounding
- Completely automated

 Invocation done similarly
Conclusions

• Approaches structured differently
  – DIANE framework matchmaking centered
  – Glue discovery alone much leaner approach, Glue+WebRatio similar scope

• Quite different approach to matchmaking
  – DIANE: set based matching, generic matching rules, fuzzy sets
  – Glue: domain specific rule-based matching of goal instance vs. web service instances
  – Advantages / Disadvantages depend on use case

• Usually workarounds for the inferior approach (see evaluation results)
Pros and Cons, Hard and Easy Things

- + rules
- - preferences & ranking
- + standard reasoning tools, WSMO compliant
- - does not support offer configuration
- + higher expressivity of ontology language
- - domain specific matching rules

- - rules
- + preferences & ranking
- - non-standard reasoning and modeling
- + supports offer configuration
- - limited expressivity of ontology language
- + matching rules generic (advantageous for complex scenarios)
Thank you!

Questions?

Ulrich Küster (Ulrich.Kuester@uni-jena.de)
Andrea Turati (Andrea.Turati@cefriel.it)